

AUREA Design System Integration Guide

For Claude Code Framework Build

This document guides Claude Code in implementing the complete Aurea design system for the ingredient search feature.

Overview

You have three design deliverables:

- AUREA_UI_Design_System.md** - Complete design specifications (17 sections)
- IngredientSearch.jsx** - Interactive React component prototype
- design-tokens.css** - Global CSS variables and utilities

Design Integration Workflow

Step 1: Global Design Setup

What Claude Code should do:

frontend/src/styles/

- |— design-tokens.css ← Copy this file as-is
- |— globals.css ← Import design-tokens.css
- |— animations.css ← Define @keyframes
- |— layout.css ← Page layout styles

Key:

- Import design tokens first: `@import 'design-tokens.css';`
- All variables available globally via CSS custom properties
- Example usage: `color: var(--color-brand-green);`

Step 2: Component Structure

Claude Code should build:

```
frontend/src/components/
├── IngredientSearch/
│   ├── IngredientSearch.jsx    ← Main container
│   ├── SearchBar.jsx          ← Search input
│   ├── IngredientCard.jsx     ← Main circular card
│   ├── HealthBreakdown.jsx    ← Category sections
│   ├── ScoreBar.jsx           ← Progress indicator
│   └── styles/
│       ├── IngredientSearch.module.css
│       ├── SearchBar.module.css
│       ├── IngredientCard.module.css
│       ├── HealthBreakdown.module.css
│       └── ScoreBar.module.css
├──
├── Navigation/
│   ├── Navigation.jsx
│   └── Navigation.module.css
├──
└── common/
    ├── Button.jsx             ← Reusable button
    ├── LoadingSpinner.jsx     ← Spinner component
    ├── ErrorCard.jsx          ← Error state
    └── Toast.jsx               ← Notifications
```

Design Token Reference

Colors - Quick Lookup

Primary Brand:

```
css

--color-brand-green: #2D7D5E    /* Main action color */
--color-brand-green-dark: #246047 /* Hover state */
--color-brand-green-light: #E0F0EC /* Light background */
```

Navigation:

```
css

--color-warm-beige: #E8DCC8      /* Nav pills */
--color-cream: #F5F3F0          /* Page background */
```

Score Indicators:

```
css

--color-score-excellent: #10B981 /* 90-100 */
--color-score-good: #84CC16      /* 70-89 */
--color-score-moderate: #F59E0B  /* 50-69 */
--color-score-concerning: #EF9B54 /* 30-49 */
--color-score-high-risk: #DC2626 /* 0-29 */
```

Score Bar Gradient:

CSS

```
--color-bar-1: #F87171 (Red)
--color-bar-2: #FB923C (Orange)
--color-bar-3: #FBBF24 (Amber)
--color-bar-4: #A3E635 (Lime)
--color-bar-5: #34D399 (Green)
```

Typography

Fonts:

CSS

```
--font-display: 'Garamond', 'Georgia', serif /* Headings */
--font-body: 'Inter', '-apple-system', sans-serif /* Body */
```

Sizes:

CSS

```
--font-size-h1: 48px
--font-size-h2: 32px
--font-size-h3: 24px
--font-size-body-lg: 18px
--font-size-body: 16px
--font-size-body-sm: 14px
--font-size-caption: 12px
```

Weights:

CSS

```
--font-weight-normal: 400
--font-weight-medium: 500
--font-weight-semibold: 600
--font-weight-bold: 700
```

Spacing

Scale (multiples of 4px):

CSS

```
--space-xs: 4px
--space-sm: 8px
--space-md: 12px
--space-lg: 16px
--space-xl: 24px
--space-2xl: 32px
--space-3xl: 40px
--space-4xl: 48px
--space-5xl: 64px
```

Border Radius

CSS

```
--radius-sm: 4px      /* Small elements */
--radius-md: 8px      /* Cards, buttons */
--radius-lg: 12px     /* Larger sections */
--radius-xl: 16px     /* Modal containers */
--radius-full: 9999px /* Circles, pills */
```

Shadows

```
css

--shadow-sm: 0 1px 2px rgba(0, 0, 0, 0.05)  /* Subtle */
--shadow-md: 0 4px 6px rgba(0, 0, 0, 0.07)  /* Medium */
--shadow-lg: 0 10px 15px rgba(0, 0, 0, 0.1) /* Large */
--shadow-xl: 0 20px 25px rgba(0, 0, 0, 0.1) /* XL */
```

Transitions

```
css

--transition-quick: 100ms cubic-bezier(0.4, 0, 0.2, 1)
--transition: 200ms cubic-bezier(0.4, 0, 0.2, 1)
--transition-slow: 300ms cubic-bezier(0.34, 1.56, 0.64, 1)
```

Component Specifications

1. Navigation Component

File: `Navigation.jsx`

Specifications from Design System (Section 3):

- Top navigation bar with pill-style buttons
- Navigation items: History, Insights, Library, About Us
- Profile icon on the right
- Background: `var(--color-cream)`
- Height: 56px (mobile), 64px (desktop)
- Pills: `var(--color-warm-beige)` background, `var(--radius-full)` border-radius

Props:

```
jsx

<Navigation
  activeTab="search"
  onTabChange={(tab) => {}}
/>
```

2. SearchBar Component

File: `SearchBar.jsx`

Specifications (Section 4):

- Rounded search input with microphone icon
- Autocomplete dropdown below
- Border: `2px solid var(--color-border)`, `var(--radius-full)`
- Focus: `var(--color-brand-green)` border, 3px shadow
- Placeholder color: `var(--color-text-muted)`

Props:

```
jsx<SearchBar
  value={searchQuery}
  onChange={handleSearch}
  onSelect={handleSelect}
  suggestions={autocompleteList}
  isLoading={false}
/>
```

3. IngredientCard Component

File: `IngredientCard.jsx`

Specifications (Section 5):

- Circular container (aspect-ratio: 1/1, border-radius: 50%)
- Background: `var(--color-cream)`
- Border: `3px solid #D4C5B0`
- Center content vertically
- Score number color: Dynamic based on score (use `getScoreColor` function)

Props:

```
jsx<IngredientCard
  ingredient={{
    name: "Sugar",
    overallScore: 85,
    riskLevel: "Low Concern",
    description: "...",
    categories: {...}
  }}
  isLoading={false}
/>
```

4. ScoreBar Component

File: `ScoreBar.jsx`

Specifications (Section 6):

- Height: 8px

- Background: Gradient (linear-gradient(90deg, ...))
- Animated fill based on score
- Indicator line at exact score position

Props:

```
jsx
<ScoreBar
  score={85}
  animated={true}
/>
```

5. HealthBreakdown Component

File: `HealthBreakdown.jsx`

Specifications (Section 6):

- 5 category sections:
 1. Blood Sugar Impact
 2. Inflammation Potential
 3. Gut Impact
 4. Hormonal Relevance
 5. Evidence Confidence
- Each section has:
 - Title with bottom border
 - Description
 - Details list (with bullets)
 - Progress bar (except Evidence)

Props:

```
jsx
<HealthBreakdown
  categories={ingredientData.categories}
  animated={true}
/>
```

6. Button Component

File: `Button.jsx`

Specifications (Section 7):

- Two variants: primary, secondary
- Primary: `var(--color-brand-green)` bg, white text
- Secondary: white bg, `var(--color-brand-green)` border
- Disabled state: 60% opacity, not-allowed cursor

- Hover: darker color + shadow for primary

Props:

```
jsx<Button  variant="primary" // or "secondary"  onClick={handleClick}  disabled={false}  loading={false}>  ADD TO TODAY'S LOG</Button>
```

7. LoadingSpinner Component

File: LoadingSpinner.jsx

Specifications (Section 8):

- Circular spinner animation
- Diameter: 40px
- Color: var(--color-brand-green)
- Rotation animation, 1s duration
- Center in container

Props:

```
jsx<LoadingSpinner size="md" color="primary" />
```

8. ErrorCard Component

File: ErrorCard.jsx

Specifications (Section 9):

- Background: var(--color-error-bg) (#FEF3C7)
- Border: 2px solid var(--color-warning)
- Padding: var(--space-xl)
- Icon, title, description, action button

Props:

```
jsx
```

```
<ErrorCard
  title="Ingredient Not Found"
  description="Try searching for something else"
  action={{
    label: "CLEAR SEARCH",
    onClick: handleClear
  }}
/>
```

9. Toast Component

File: `Toast.jsx`

Specifications:

- Position: bottom-right
- Background: `var(--color-success)` (green)
- Text: white
- Auto-dismiss: 3 seconds

Props:

```
jsx

<Toast
  message="Saved to today's log!"
  type="success" // success, error, warning
  autoClose={3000}
/>
```

CSS Module Pattern

Each component should have a CSS module:

CSS


```
/* IngredientCard.module.css */

.container {
  display: flex;
  flex-direction: column;
  align-items: center;
  width: 100%;
  max-width: 500px;
  aspect-ratio: 1 / 1;
  background-color: var(--color-cream);
  border: 3px solid #D4C5B0;
  border-radius: var(--radius-full);
  padding: var(--space-4xl);
  box-shadow: var(--shadow-lg);
  margin-bottom: var(--space-xl);
}

.scoreNumber {
  font-family: var(--font-display);
  font-size: 72px;
  font-weight: var(--font-weight-bold);
  margin-bottom: var(--space-xl);
}

.scoreNumber.excellent {
  color: var(--color-score-excellent);
}

.scoreNumber.good {
  color: var(--color-score-good);
}

/* ... etc for other states ... */
```

Color Score Mapping Function

Claude Code should implement this utility:

```
javascript
```

```
// utils/scoring.js

export const getScoreColor = (score) => {
  if (score >= 90) return 'var(--color-score-excellent)';
  if (score >= 70) return 'var(--color-score-good)';
  if (score >= 50) return 'var(--color-score-moderate)';
  if (score >= 30) return 'var(--color-score-concerning)';
  return 'var(--color-score-high-risk)';
};

export const getScoreLabel = (score) => {
  if (score >= 90) return 'Excellent';
  if (score >= 70) return 'Good';
  if (score >= 50) return 'Moderate';
  if (score >= 30) return 'Concerning';
  return 'High Risk';
};

export const getRiskLevel = (score) => {
  if (score >= 70) return 'Low Concern';
  if (score >= 50) return 'Moderate Concern';
  return 'High Concern';
};
```

Responsive Breakpoints

Mobile-First Approach:

```
css

/* Base styles (mobile) */
.container {
  width: 100%;
  padding: var(--space-lg);
}

/* Tablet */
@media (min-width: 640px) {
  .container {
    padding: var(--space-xl);
    max-width: 600px;
  }
}

/* Desktop */
@media (min-width: 1025px) {
  .container {
    max-width: 800px;
    padding: var(--space-2xl);
  }
}
```

Animation Specifications

From Design System (Section 13):

Entrance Animations

css

```
@keyframes fadeIn {
  from { opacity: 0; }
  to { opacity: 1; }
}

@keyframes slideUp {
  from {
    transform: translateY(20px);
    opacity: 0;
  }
  to {
    transform: translateY(0);
    opacity: 1;
  }
}

@keyframes scaleIn {
  from {
    transform: scale(0.95);
    opacity: 0;
  }
  to {
    transform: scale(1);
    opacity: 1;
  }
}
```

Usage:

jsx

```
// Card entrance
<IngredientCard style={{
  animation: 'slideUp 300ms cubic-bezier(0.34, 1.56, 0.64, 1)'
}} />

// Category staggered entrance
{categories.map((cat, idx) => (
  <div
    key={cat.id}
    style={{
      animation: `fadeIn 200ms ease-in ${idx * 100}ms both`
    }}
  >
    ...
  </div>
))}
```

Accessibility Requirements

From Design System (Section 12):

Color Contrast

- All text meets WCAG AA: 4.5:1 for normal text, 3:1 for large text
- No information conveyed by color alone

Keyboard Navigation

- Tab order: Navigation → Search → Results
- Focus indicators: 2px outline
- Escape key closes dropdowns

Screen Readers

- All icons have `aria-label`
- Headings properly hierarchical
- Form labels associated with inputs
- Progress bars have `role="progressbar"`

Example:

```
jsx<div
  role="progressbar"
  aria-valuenow={score}
  aria-valuemin={0}
  aria-valuemax={100}
  aria-label="Blood sugar impact score"
>
  <div className={styles.scoreBar} />
</div>
```

Integration Checklist

Claude Code should complete:

- ☐ Import `design-tokens.css` in `globals.css`
- ☐ Create all component files with CSS modules
- ☐ Implement color score mapping functions
- ☐ Add hover/focus/active states to all interactive elements
- ☐ Implement all animations with proper timing
- ☐ Test responsive design at 3 breakpoints (mobile, tablet, desktop)
- ☐ Verify accessibility (keyboard nav, screen reader, contrast)
- ☐ Add loading states to all async operations
- ☐ Implement error states with ErrorCard component
- ☐ Add Toast notifications for user feedback

File Deliverables

Claude Code will create:

```
frontend/src/
├── styles/
│   ├── design-tokens.css ← USE PROVIDED
│   ├── globals.css
│   ├── animations.css
│   └── layout.css
│
├── components/
│   ├── IngredientSearch/
│   │   ├── IngredientSearch.jsx
│   │   ├── SearchBar.jsx
│   │   ├── IngredientCard.jsx
│   │   ├── HealthBreakdown.jsx
│   │   ├── ScoreBar.jsx
│   │   └── styles/
│   │       └── *.module.css
│   │
│   ├── Navigation/
│   │   ├── Navigation.jsx
│   │   └── Navigation.module.css
│   │
│   └── common/
│       ├── Button.jsx
│       ├── LoadingSpinner.jsx
│       ├── ErrorCard.jsx
│       └── Toast.jsx
│
├── hooks/
│   └── useIngredientSearch.js
│
├── utils/
│   └── scoring.js
│
└── pages/
    └── SearchPage.jsx
```

Next Steps

1. **Claude Code receives this guide + design system**
2. **Builds all components with design tokens**
3. **Implements responsive design**
4. **Tests accessibility**
5. **Delivers production-ready component library**

Status: Ready for Implementation

Next Review: After component build complete